# **Chainlit-Based Text-to-SQL System**

## **Overview**

This project provides a **Chainlit-based conversational interface** for generating SQL queries from user queries using **CSV and PDF files**. It integrates **FAISS for indexing** and includes dynamic chart generation.

## **Features**

* **File Upload Handling**: Supports CSV and PDF file uploads.
* **SQL Query Generation**: Uses an AI-based system to generate SQL queries from natural language.
* **FAISS Indexing**: Enables efficient document search.
* **Dynamic Pie Chart Generation**: Displays a pie chart based on query results.

## **Installation**

Ensure you have Python installed (preferably 3.8+). Then, install dependencies:

pip install chainlit faiss-cpu pandas json

Clone the repository and navigate into the project directory:

git clone <repository-url>

cd <project-directory>

## **Usage**

Run the Chainlit app:

chainlit run app.py

Upload **both a CSV and a PDF file**, then ask SQL-related queries.

## **Code Structure**

### **1. File Upload Handling**

csv\_path = None

pdf\_path = None

text2sql = None

* Stores uploaded file paths.
* Ensures both CSV and PDF are uploaded before processing queries.

### **2. Handling User Queries**

@cl.on\_message

async def handle\_message(message: cl.Message):

* Handles file uploads and processes user queries.
* Calls generate\_sql\_query(user\_query) to fetch SQL statements.

### **3. SQL Query Generation**

response = text2sql.generate\_sql\_query(user\_query)

out = json.loads(response)

Answer = out.get('Answer', 'No answer provided')

SQL = out.get('SQL Query', None)

* Converts user query to SQL.
* Extracts the answer and generates SQL.

### **4. Dynamic Pie Chart Handling**

file\_path = "dynamic\_pie\_chart.png"

if os.path.exists(file\_path):

await cl.Message(

content="📊 \*\*Generated Pie Chart:\*\*",

elements=[cl.Image(path=file\_path)]

).send()

* Checks for the existence of a pie chart and displays it.

## **Future Enhancements**

* Add support for multiple PDFs and CSVs.
* Improve FAISS indexing for better retrieval.
* Enhance visualization features.
* Optimize SQL generation accuracy.

## 

## **Running the Application**

To start the Chainlit app, run:

cl.run()

Now, upload a **CSV and PDF**, then ask your query! 🚀